T35638

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 313 097 A2**

(12) **EUROPEAN PATENT APPLICATION**

(21) Application number: 88117683.8

(22) Date of filing: 24.10.88

(51) Int. Cl.⁴: G06F 9/38

(30) Priority: 22.10.87 JP 267796/87

(43) Date of publication of application:
26.04.89 Bulletin 89/17

(84) Designated Contracting States:
DE FR GB

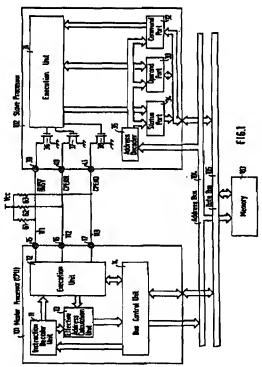
(71) Applicant: NEC CORPORATION
33-1, Shiba 5-chome, Minato-ku
Tokyo 108(JP)

(72) Inventor: Iwasaki, Junichi
c/o NEC Corporation 33-1, Shiba 5-chome
Minato-ku Tokyo(JP)
Inventor: Kojima, Shingo
c/o NEC Corporation 33-1, Shiba 5-chome
Minato-ku Tokyo(JP)

(74) Representative: Glawe, Delfs, Moll & Partner
Patentanwälte
Postfach 26 01 62 Liebherrstrasse 20
D-8000 München 26(DE)

(54) **Microcomputer system including master processor and slave processor.**

(57) A microcomputer system including a master processor and a coprocessor interconnected via a bus is disclosed. The coprocessor supplies first, second and third signals to the master processor, the first signal taking an active level when the coprocessor requests a wait condition of the master processor, the second signal taking an active level at when the first signal is changed to the inactive level in a state of occurrence of arithmetic exception in the coprocessor, and the third signal taking an active level when the coprocessor is free of execution of an instruction and of occurrence of the arithmetic exception. When the coprocessor executes an instruction whose execution result is not required to be returned to the master processor, the master processor detects the level of the third signal and if the third signal is at the active level, the master processor starts to execute subsequent instructions irrespective of the first and second signals, in parallel to the execution of the instruction by the coprocessor.



EP 0 313 097 A2

Microcomputer System Including Master Processor and Slave Processor

CROSS REFERENCE TO RELATED APPLICATION

This application is related to the following copending application:

Ser. No. 119,601, filed by the same assignee for Takashi Nakayama on Nov. 12, 1987, under a title of "Micro Processor Capable of Being Connected with Coprocessor".

BACKGROUND OF THE INVENTION

The present invention relates to a microcomputer system and, more particularly, to a microcomputer system wherein a master processor and at least one coprocessor are interconnected via a bus.

In a microprocessor fabricated on a single semiconductor chip, a very large number of circuit elements are required but the number of circuit elements capable of being formed in the single chip is limited. Therefore, it is difficult to fabricate such a unit in a single chip that executes high level instructions such as a floating-point arithmetic operation, a function arithmetic operation, etc., at a high speed. In order to solve this problem, a coprocessor is employed, which executes the high level instructions in place of the microprocessor. The coprocessor operates under the control of the microprocessor and is thus called a "slave processor". The microprocessor operates by itself as a central processing unit (CPU) to control the coprocessor along with a memory and peripheral units and is thus called a "master processor".

When the master processor decodes the high level instruction, it makes access to the slave processor to transfer the high level instruction as well as one or more operands, if necessary. The slave processor is thereby brought into an operative condition to execute the high level instruction thus supplied. If the master processor is not connected with slave processor, it should execute the high level instructions by use of its own arithmetic unit. Therefore, the master processor is required to detect whether or not the slave processor is connected thereto. In a case of provision of the slave processor, moreover, if the arithmetic exception occurs in the arithmetic result or arithmetic procedure of the slave processor, the master processor should perform the processing operation for the arithmetic exception occurring in the slave processor. Therefore, the master processor is further required to detect whether or not the status information of the slave processor is necessary to be read therefrom.

For these purposes, the master processor includes first and second terminals supplied respectively with a busy signal and a status read request signal from the slave processor, as disclosed in the above copending application. When the master processor decodes and transfers the high level instruction to the slave processor, the slave processor changes the busy signal from an inactive level to an active level and holds the active level of the busy signal during the executing period thereof. When the slave processor completes the execution of the high level instruction, it changes the busy signal from the active level to the inactive level. At this time, in a case of no occurrence of the arithmetic exception, the slave processor produces an inactive level of the status read request signal when it changes the busy signal from the active level to the inactive level. In contrast, if the arithmetic exception occurs, the slave processor produces an active level of the status read request signal. On the other hand, the master processor detects the level of the first terminal, i.e. the level of the busy signal, and further detects the level of the status read request signal supplied to the second terminal when the busy signal is in the inactive level. Thus, if the slave processor is not provided, both the first and second terminals of the master processor, i.e. both the busy signal and status read request signal, are maintained to the inactive level and the active level, respectively. On the other hand, in cases of provision of the slave processor and no occurrence of the arithmetic exception, the status read request signal is in the inactive level when the busy signal is changed from the active level to the inactive level. As a result, the master processor can detect whether or not the slave processor is connected thereto and whether or not the arithmetic exception occurs in the slave processor.

The high level instructions to be executed by the slave processor are divided into two types: the first type of instructions is such that the execution result is required to be returned to the master processor and the second type of instructions is such that the execution result is not required to be returned to the master processor and is stored into, for example, a register of the slave processor. When the first type instruction is read from an instruction memory, the master processor must wait for the completion of the execution of this type instruction by the slave processor. On the other hand, when the slave processor executes the second type high level instruction, the master proces-

sor can execute subsequent low level instructions in parallel to the execution of the second type instruction by the slave processor.

Although the master processor does not require the execution result of the second type high level instruction, it shoud detect whether or not the arithmetic exception occurs in the slave processor and, if occurs, read the status information to detect the kind of the arithmetic exception. To this end, the master processor always detects the levels at the first and second terminals, i.e. the levels of the busy signal and the status read request signal, after transferring the high level instruction irrespective of the types of the high level instruction which is now to be executed by the slave processor. When the slave processor receives a new high level instruction, if the arithmetic exception has occurred in the execution result or the executing procedure of the precedingly supplied second type instruction, the slave processor changes the busy signal to the active level and immediately returns it to the inactive level with holding the status read request signal at the active level. In a case where the arithmetic exception has not occurs, the slave processor changes the busy signal to the active level and after changing the status read request signal to the inactive level, returns the busy signal to the inactive level. Thus, the master processor can detect whether or not the arithmetic exception has occurs in the execution of the second type high level instruction by the slave processor. However, since the master processor detects the levels of the first and second terminals whenever the high level instruction is decoded, it is delayed to start execution of subsequent low level instructions when a new one of the second type high level instructions is decoded and when the arithmetic exception has not occurred in the execution of the preceding one of second type high level instructions.

SUMMARY OF THE INVENTION

Therefore, a primary object of the present invention is to provide an improved microcomputer system in which a master processor and at least one slave processor are interconnected via a bus.

Another object of the present invention is to provide a microcomputer system in which a master processor can execute at least one low level instruction in parallel to the execution of a high level instruction by a slave processor without delaying the starting time of the execution of the low level instruction.

Still another object of the present invention is to provide an improved master processor or a microprocessor capable of being connected with at least one coprocessor and executing at least one instruction while the coprocessor is executing another instruction.

A microcomputer system according to the present invention comprises a master processor, a coprocessor, and a bus interconnecting the master processor and the coprocessor, and the master processor includes first, second and third terminals supplied respectively with first, second and third signals from the coprocessor. The first signal corresponds to the above-mentioned busy signal and thus takes an active level when the master processor is required to be in a wait condition, and the second signal corresponds to the above-mentioned status read request signal and thus takes an active level at least when the first signal is changed from the active level to an inactive level in a state of occurrence of arithmetic exception. The third signal is provided in accordance with the present invention and takes an active level when the coprocessor is not be executing an instruction and when arithmetic exception does not occur in the execution of a preceding instruction. The master processor further includes means coupled to the third terminal for detecting the level of the third signal, means responsive to a first output of the detecting means representative of the active level of the third signal for executing at least one instruction in parallel to the execution of the coprocessor irrespective of the levels of the first and second terminals, and means responsive to a second output of the detecting means representative of the inactive level of the third signal for bringing the master processor into the wait condition until the first signal is changed to the inactive level and for performing a subsequent processing operation responsive to the second signal after the first signal is changed to the inactive level.

Thus, when the second type high level instruction is read out and the third signal takes the active level at this time, the master processor can start the execution of at least one subsequent instruction in parallel to the execution of that second type high level instruction by the coprocessor without being brought into the wait condition by the first signal. The executing start timing of the subsequent instruction is thereby fastened.

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other objects, advantages and features of the present invention will be more apparent from the following description taken in conjunction with the accompany drawings, in which:

Fig. 1 is a block diagram representative of an embodiment of the present invention;

Fig. 2 is a flow chart representative of a sequence control procedure of a master processor shown in Fig. 1;

Figs. 3, 4, 5, 6 and 7 are timing chart each representative of operations of a microcomputer system shown in Fig. 1;

Fig. 8 is a block diagram representative of another embodiment of the present invention; and

Fig. 9 is a timing chart representative of operations of a microcomputer system shown in Fig. 8.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring to Fig. 1, a microcomputer system according to an embodiment of the present invention includes a master processor 101 as a CPU, a slave processor 102 as a coprocessor and a memory 103 which are interconnected via an address bus 104 and a data Bus 105. The slave processor 102 operates under the control of the master processor 101 and executes at a high speed high level instructions such as a floating-point arithmetic operation, a function arithmetic operation, etc., in place of the master processor 101. The memory 103 stores data and a program including low level instructions and high level instructions and is made access by the master processor 101.

The master processor 101 includes an instruction decoder unit 11 decoding instructions which are executed by the master and slave processors 101 and 102, an execution unit 12 controlling the whole operation of the master processor 101 and executing the low level instructions under the sequential control of a microprogram, an effective address calculation unit 13 calculating an effective address of each operand for the respective instructions, and a bus control unit 14 controlling the address and data buses 104 and 105 to make access to the memory 103 and the slave processor 102. These units 11 to 14 are coupled as shown via internal buses. The master processor 101 further includes a first input terminal 15 supplied with a BUSY signal, a second input terminal 16 supplied with a CPERR signal and a third input terminal 17 supplied with a CPEND signal, the third input terminal 17 being provided in accordance with the present invention.

The slave processor 102 includes an execution unit 31 performing an arithmetic operation responsive to each of the high level instructions, a command port 32 receiving and temporarily storing command code (opecodes) of each instruction transferred via the data bus 105, a status port 33 temporarily storing the status information supplied from the execution unit 31 and outputting it onto the data bus 105, an operand port 34 temporarily storing one or more operands transferred via the data bus 105 for the arithmetic operation of the unit 31 and receiving and outputting the execution result from the unit 31 onto the data bus 105, if necessary. The slave processor 102 also includes and address decoder 35 decoding address information supplied via the address bus 104 and selecting one of the units 32, 33 and 34 in response thereto to write or read data into or from the selected unit. These units 31 to 35 are coupled as shown via internal buses. The slave processor 102 further includes three output terminals 39, 40 and 41 and three transistors 36, 37 and 38 of an open-drain type. Each of the transistors 38, 37 and 38 has a drain-source path connected between each of the terminals 39, 40 and 41 and a ground point and a gate connected the execution unit 31. The terminal 41 and transistor 38 are provided in accordance with the present invention.

The output terminals 39, 40 and 41 of the slave processor 102 are connected to the input terminals 15, 16 and 17 of the master processor 101 via signal lines 111, 112 and 113, respectively, which lines are in turn pulled up to a power voltage via resistors 61, 62 and 63, respectively. Accordingly, so long as the transistors 36 to 38 are in the nonconductive state, the BUSY, CPERR and CPEND signals, i.e. the input terminals 15 to 16 of the master processor 101, are maintained at the high level. When the slave processor 102 receives the first type of the high level instruction representing that the execution result is required to be returned to the master processor 101, the execution unit 31 of the slave processor 102 turns the transistor 36 ON to change the BUSY signal to an active level (low level in this embodiment) unit the execution of the receiving instruction is completed. On the other hand, when the slave processor 102 receives the second type of the high level instruction representing that the execution result is not required to be returned to the master processor 101, the execution unit 31 thereof turns the transistor 36 ON and immediately thereafter turns it OFF. Accordingly, the BUSY signal is changed to the low level and immediated returned to the high level. However, when the slave processor 102 receives the subsequent second type instruction while it is executing the current second type instruction, the execution unit 31 remains the transistor 36 in the conductive level to produce the low level BUSY signal until the completion of the execution of the current second type instruction. The execution unit 31 of the slave processor 102 turns the transistor 37 ON with changing the transistor 38 from the conductive state to the nonconductive state, when 110 arithmetic exception occurs in the execution

result or the executing procedure of the high level instruction. The CPERR signal thereby takes an inactive level (the high level). On the other hand, when the arithmetic exception occurs or when the master processor 101 does not yet read from the status port 34 the status information representative of the arithmetic exception, the transistor 37 is held in the non conductive state to produce an active level (the high level) of the CPERR signal. The execution unit 31 of the slave processor 102 turns the transistor 38 ON during the executing period of the high level instruction, and when the arithmetic exception occurs, it remains the transistor 38 in the conductive state until the master processor 101 reads from the status port 34 the status information representative of the arithmetic exception. When the slave processor 102 is free of the executing state of the high level instruction and when it stores no arithmetic exception, the transistor 38 is in the nonconductive state. The CPEND signal thereby takes an active level (the high level). Since the active and inactive levels of the BUSY, CPERR and CPEND signals are determined as described above, the master processor 101 can judge the operating state of the slave processor 102 by detecting the levels of the input terminals 15, 16 and 17.

When the high level instruction to be executed by the slave processor 102 is read from the memory 103 and then decoded by the instruction decoder unit 11 of the master processor 101, a protocol operating procedure between the master and slave processors 101 and 102 is activated in accordance with the microprogram stored in execution unit 12. This procedure is shown in Fig. 2 as a flow chart. Assuming that the read-out instruction denotes a dyadic operation on two operands stored in the memory 103, the execution unit 12 prompts the bus control unit 14 to read these operands from the memory 14 and to transfer them to the operand port 33 of the slave processor 102. Thus, the bus control unit 14 controls the address and data buses 104 and 105 to writes the operands into the operand port 33 (step 200). If the operands has been already provided in registers of the slave processor 102, the step 200 is not carried out. Thereafter, the command code (opecode) of the high level instruction is transferred to the command port 32 of the slave processor 102 via the data bus 105 under the control of the bus control unit 14 (step 201). In a subsequent step 202, it is judged which type instruction is transferred. Assuming that the instruction to be transferred or executed is the second type instruction representing that the execution result is not required to be returned to the master processor 101, e step 211 is performed.

In this step 211, the execution unit 12 of the master processor 101 detects the level of the terminal 17, i.e. the CPEND signal, not detects the levels of the terminals 15 and 16, i.e. the BUSY and CPERR signals, in accordance with the present invention. Assuming that the slave processor 102 is not in the executing state (i.e., in the standby state) and the arithmetic exception has not occurred in the execution of the preceding high level instruction, the CPEND signal, i.e. the terminal 17, takes the active level (high level), as shown in Fig. 3. Therefore, the master processor 101 waits for the completion of the bus cycle for transferring the command code to the command port 32 (step 212) and then reads and executes the subsequent instructions (step 213). That is, the master processor 101 executes the subsequent instructions in parallel to the execution of the second type high level instruction by the slave processor 102, as shown in Fig. 3. Since the level of the BUSY signal is not detected, the active level thereof is ignored. When the slave processor 102 completes the execution of the instruction without occurrence of the arithmetic exception, it stores the execution result into the internal register thereof and turns the transistor 38 OFF to produce the active level of the CPEND signal, as shown in Fig. 3.

In contrast, when the arithmetic exception occurs in the execution of the second type instruction, the execution unit 31 of the slave processor 102 maintains the transistor 38 in the conductive state to produce the inactive level (low level) of the CPEND signal, as shown in Fig. 4. The unit 31 further writes into the status port 34 the status information representative of the arithmetic exception. In this condition, when a subsequent second type high level instruction is read out of the memory and decoded by the unit 11, the master processor 101 performs the steps 200, 201, 202 and 211, but the CPEND signal takes the inactive level, so that the operating procedure shifts to a step 203. In this step 203, the master processor 101 waits for the completion of writing the command code of that subsequent second type high level instruction into the command port 32. In response thereto, the execution unit 31 turns the transistor 36 ON and immediately thereafter turns it OFF. Since the status port 34 stores the information representative of the exception, the unit 31 turns the transistor 36 OFF with remaining the transistor 37 in the nonconductive state to produce the active level of the CPERR signal, as shown in Fig. 4. On the other hand, the master processor 101 is brought into a wait condition until the BUSY signal takes the inactive level (step 204) and thereafter it detects the level of the CPERR signal (step 205). Since the CPERR signal is at the active level, the master processor 101 reads the status information out of the status port 34 of the slave processor 102 (step 206), as shown in Fig. 4. The read-out status

information is judged in a step 207, and in accordance with the value thereof, the master processor 102 performs an processing operation for the arithmetic exception (step 208) or an processing operation for the slave processor absence exception (step 209). Since the exception information stored in the port 34 has been read out, the execution unit 31 turns the transistor 38 OFF to produce the active level of the CPEND signal. During the processing operation for the exception by the master processor 101, the slave processor 102 is brought into the standby condition. After the completion of the processing operation for the exception, the master processor 101 restarts to execute the program from the suspended address.

Assume now that, as shown in Fig. 5, a subsequent second type high level instruction ② is read from the memory 103 while the slave processor 102 is executing the preceding second type high level instruction ①. In response to the reading-out of the subsequent instruction ②, the master processor 101 performs the steps 200,201 and 202 and advances to the step 211. Since the slave processor 102 is yet in the executing state of the instruction, the CPEND signal takes the inactive level, as shown in Fig. 5. Therefore, the master processor 104 waits for the end of the bus cycle (step 203) and detects the level of the BUSY signal. In response to a fact that the command code of the subsequent instruction ② is written into the command port 32, the slave processor 102 turns the transistor 38 ON, and in this case the execution unit 31 is in the executing state of the instruction ① , so that the transistor 36 is held in the conductive state to maintain the active level of BUSY signal until the execution of instruction① is completed, as shown in Fig. 5. As a result, the master processor 101 is brought into the wait condition (step 204). When the execution of the instruction ① is completed without arithmetic exception, the execution unit 31 turns the transistor 36 OFF with turning the transistor 37 ON. The execution unit further starts to execute the instruction②, so that the transistor 38 is held in the conductive state. In response to the change of the BUSY signal to the inactive level, on the other hand, the master processor 101 detects the inactive level of the CPERR signal (step 205). It should be noted that the instruction ① is the second type instruction. Therefore, the master processor 101 executes the subsequent instructions (step 213).

When the first type high level instruction is read from the memory 103 and decoded by the unit 11, the master processor 101 performs the steps 200, 201, 202 and 203. The step 211 is not performed and the CPEND signal is ignored. On the other hand, the slave processor 102 maintains the BUSY signal at the active level until completion of this first type instruction, as shown in Fig. 6. During this period, the master processor 101 is brought into the wait condition (step 204). When the execution of the first type instruction is completed without occurrence of the arithmetic exception, the slave processor 102 changes the BUSY signal to the inactive level with producing the inactive level of the CPERR signal, as shown in Fig. 6. The slave processor 102 further writes the execution result into the operand port 33 thereof. The master processor 101 responds to the inactive level of the CPERR signal (step 205) and makes access to the operand port 33 to read the execution result therefrom (step 209).

In contrast, when the arithmetic exception occurs in the execution of the first type instruction, the slave processor 104 changes the BUSY signal to the inactive level with mainting the active level of the CPERR signal, as shown in Fig. 7. Accordingly, the master processor 101 makes access to the status port 34 to read the status information therefrom (step 206), judges the status (step 207) and performs the processing operation for the exception (step 208 or 210), as shown in Fig. 7.

Referring to Fig. 8, a microcomputer system according to another embodiment of the present invention includes three slave processors (coprocessors) 102-1, 102-2 and 102-3. In this drawing, the same constituents as those shown in Fig. 1 are denoted by the same reference numerals to omit further description thereof. Each of the slave processors 102-1, 102-2 and 102-3 drives the signal lines 111, 112 and 113 by use of open-drain transistors. Accordingly, when any one of the slave processors 102-1, 102-2 and 102-3 drives the signal lines 111, 112 and 113 to the low level, the master processor 101 receives the low levels of the BUSY, CPERR and CPEND signals.

Circuit operation of the system shown in Fig. 8 will be described below with reference to Fig. 9. At a time point, t_1, all the slave processors 102-1, 102-2 and 102-3 are in the standby condition and stores no arithmetic exception, so that the CPEND signal takes the active level (high level). The master processor 101 transfers a commend code of the second type instruction to the first slave processor 102-1 at a time point T_3 and starts to execute subsequent instructions at a time point T_4. When the second type high level instruction to be executed by the second slave processor 102-2 is decoded and transferred to thereto at a time point t_5, the master processor 101 detects the level of the CPEND signal at a time point t_6. Since the CPEND signal takes the inactive level (low level) at this time, the master processor 101 waits for the completion of the bus cycle (at a time point t_7) and detects the levels of the BUSY and CPERR signals (at time points t_8 and t_9). Since the second type

instruction is transferred to the second slave processor 102-2, the master processor 101 starts to executes subsequent instructions at the time point t_9.

Assume here that the arithmetic exception occurs in the first slave processor 102-1 at a time point t_{10}. However, this exception is not read out immediately, but is deferred to be informed to the master processor 101 until the processor 101 makes access again to the first slave processor 102-1. At a time point t_{11}, the master processor 101 makes again access to the first slave processor 102 to transfer the first or second type instruction. The master processor 101 detects the inactive level of the CPEND signal at a time point t_{12}, so that it waits for the completion of the bus cycle (a time point t_{13}) and for the change of the BUSY signal to the inactive level (a time point t_{14}). Since the first slave processor 102-1 holds the arithmetic exception, it changes the BUSY signal to the inactive level with holding the active level of the CPERR signal (a time point t_{14}). Therefore, the master processor 101 reads the exception status information from the status port of the first slave processor 102-1 and performs the processing operation for the exception.

The present invention is not limited to the above embodiments, but may be modified and changed without departing from the scope and spirit of the invention.

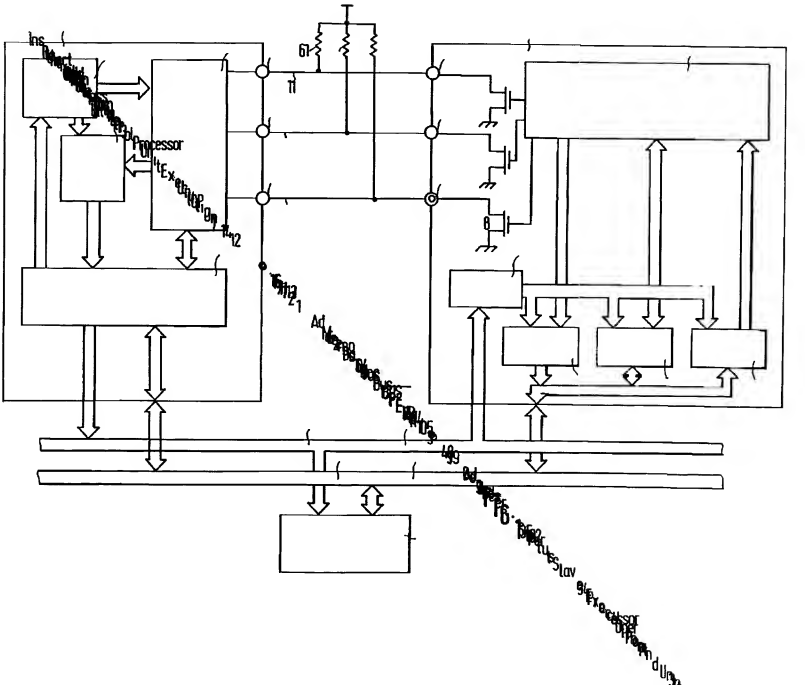
Claims

1. A processing system comprising a master processor, a coprocessor, and a bus interconnecting said master processor and said coprocessor, said coprocessor including first means for producing a first signal which takes an active level when said coprocessor requests a wait condition of said master processor, second means for producing a second signal which takes an active level at least when said first signal is changed from the active level to an inactive level in a state of occurrence of arithmetic exception in said coprocessor, and third means for producing a third signal which takes an active level when said coprocessor is free of an execution of an instruction and of the arithmetic exception, said master processor including a first terminal supplied with said first signal, a second terminal supplied with said second signal, a third terminal supplied with said third signal, fourth means coupled to said third terminal for detecting a level of said third signal, fifth means responsive to the detection result of said fourth means representing that said third signal is at the active level for executing at least one instruction simultaneously with the execution of said coprocessor irrespective of the levels of said first and second signals, and sixth means responsive to the detection result of said fourth means representing that said third signal is at the inactive level for bringing said master processor into the waite condition until said first signal is changed to the inactive level and for performing a subsequent operation responsive to the level of said second signal after said first signal is changed to the inactive level.

2. The processing system as claimed in claim 1, wherein said sixth means makes access to said coprocessor to read the execution result therefrom when said second signal is at the inactive level.

3. The processor system as claimed in claim 1, wherein said sixth means makes access to said coprocessor to read status information therefrom when said second signal is at the active level.

4. The processor system as claimed in claim 1, further comprising at least one additional coprocessor including seventh means for producing a fourth signal having the same function as said first signal, eighth means for producing a fifth signal having the same function as said second signal, and ninth means for producing a sixth signal having the same function as said third signal, said fourth, fifth and sixth signals being supplied to said first, second and third terminals of said master processor, respectively.



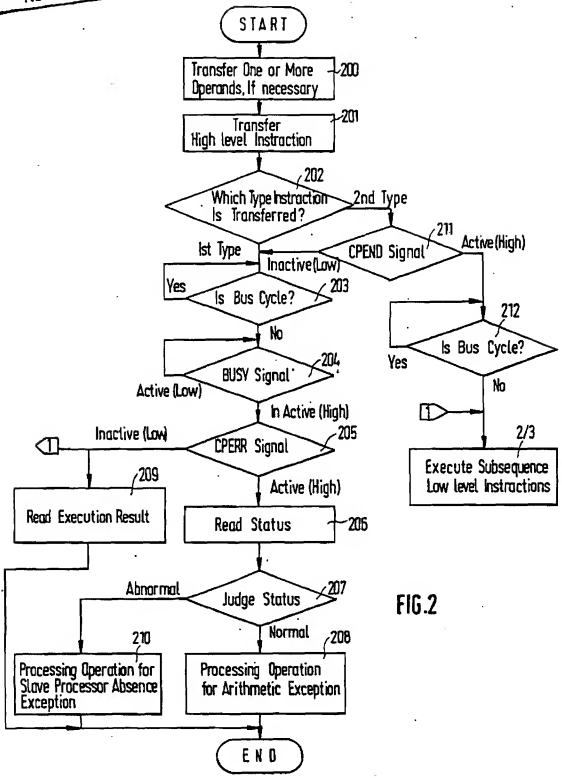
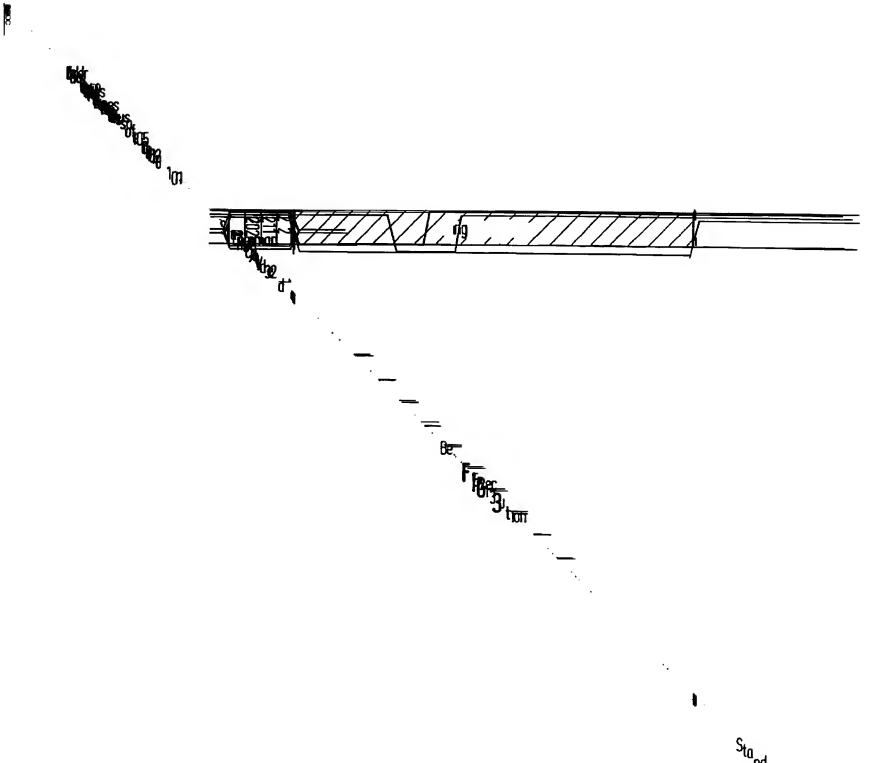


FIG.2



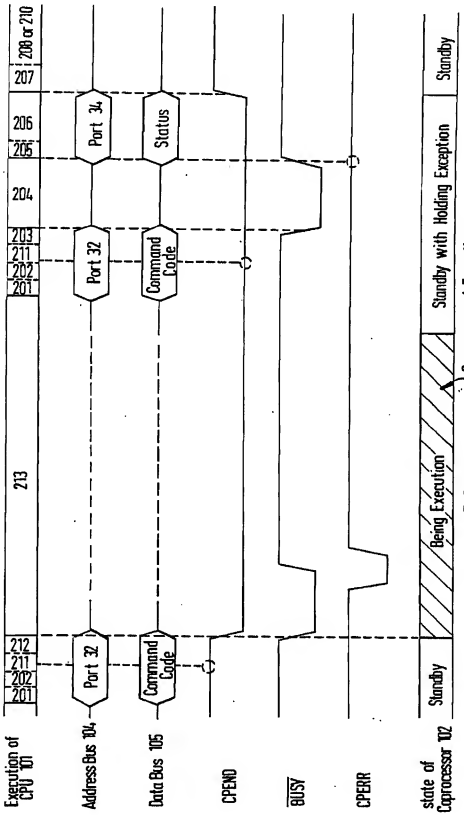
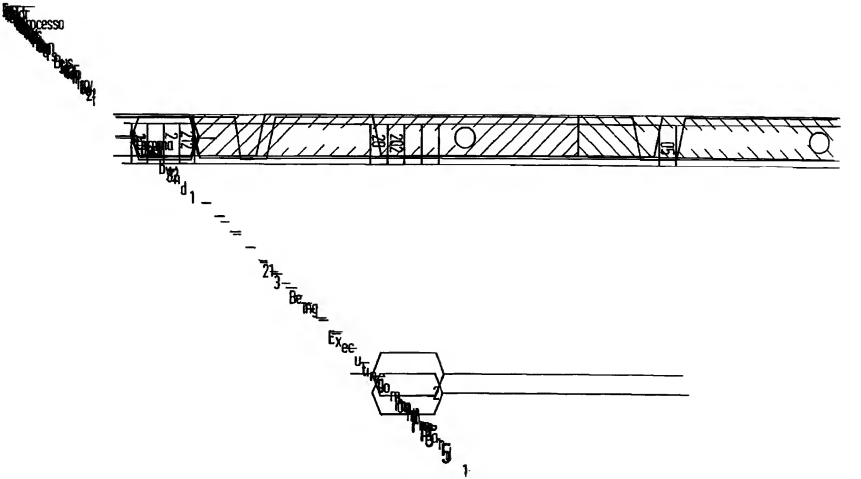
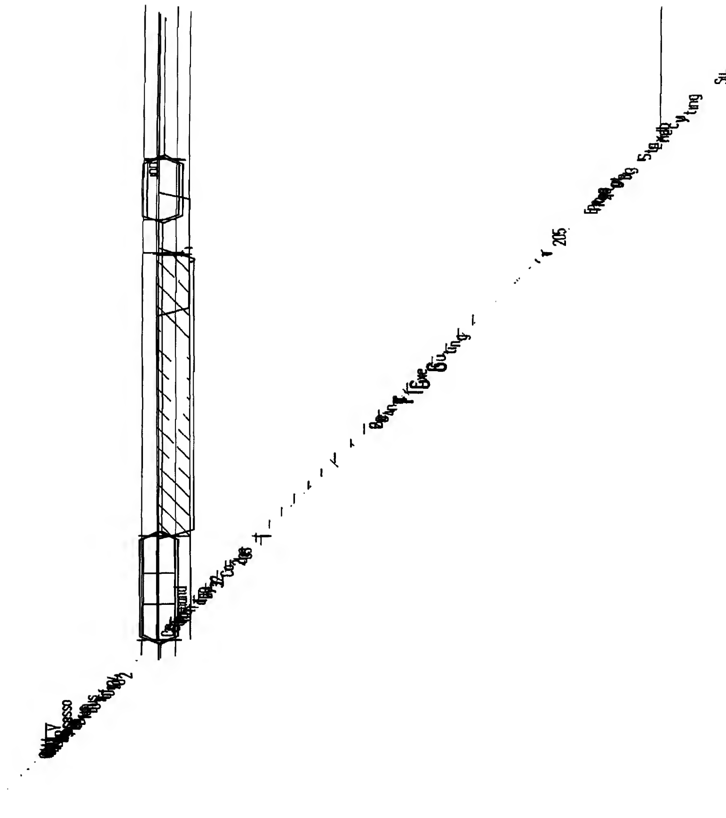
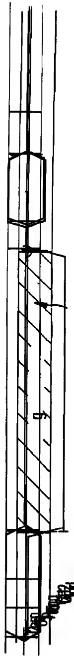
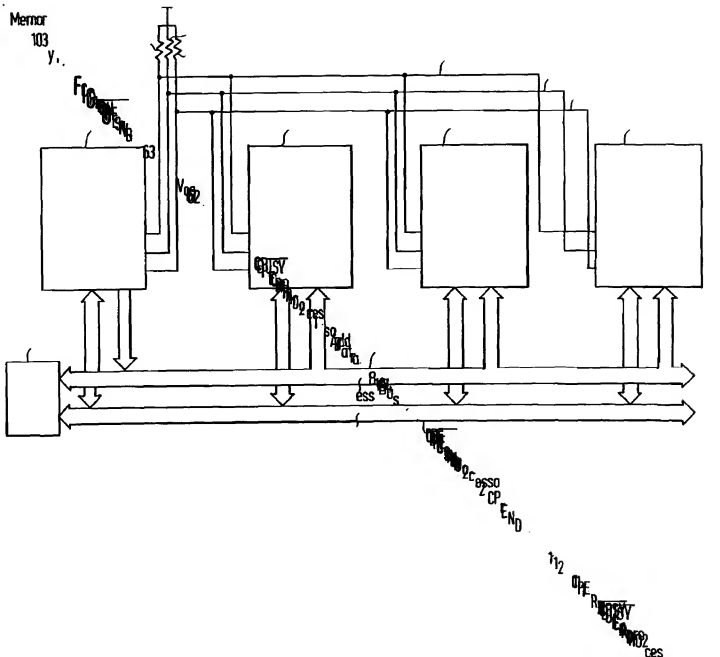


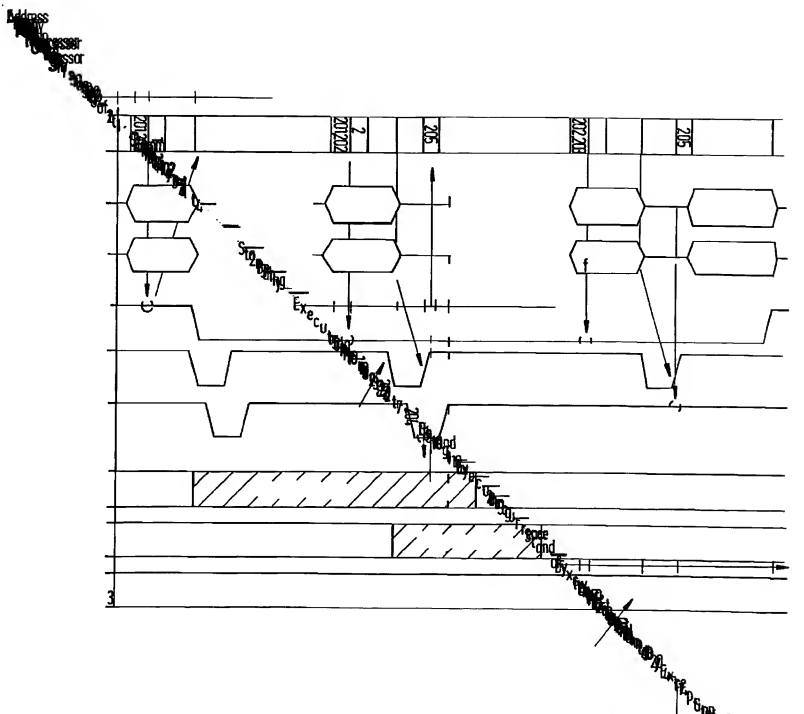
FIG.4




205




Memor
103



(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 313 097 A3**

(12) **EUROPEAN PATENT APPLICATION**

(21) Application number: **88117683.8**

(22) Date of filing: **24.10.88**

(51) Int. Cl.[5]: **G06F 9/38**

(30) Priority: **22.10.87 JP 267796/87**

(43) Date of publication of application:
26.04.89 Bulletin 89/17

(84) Designated Contracting States:
DE FR GB

(88) Date of deferred publication of the search report:
11.03.92 Bulletin 92/11

(71) Applicant: **NEC CORPORATION**
7-1, Shiba 5-chome Minato-ku
Tokyo 108-01(JP)

(72) Inventor: **Iwasaki, Junichi**
c/o NEC Corporation 33-1, Shiba 5-chome
Minato-ku Tokyo(JP)
Inventor: **Kojima, Shingo**
c/o NEC Corporation 33-1, Shiba 5-chome
Minato-ku Tokyo(JP)

(74) Representative: **Glawe, Delfs, Moll & Partner**
Patentanwälte
Postfach 26 01 62 Liebherrstrasse 20
W-8000 München 26(DE)

(54) **Microcomputer system including master processor and slave processor.**

(57) A microcomputer system including a master processor and a coprocessor interconnected via a bus is disclosed. The coprocessor supplies first, second and third signals to the master processor, the first signal taking an active level when the coprocessor requests a wait condition of the master processor, the second signal taking an active level at when the first signal is changed to the inactive level in a state of occurrence of arithmetic exception in the coprocessor, and the third signal taking an active level when the coprocessor is free of execution of an instruction and of occurrence of the arithmetic exception. When the coprocessor executes an instruction whose execution result is not required to be returned to the master processor, the master processor detects the level of the third signal and if the third signal is at the active level, the master processor starts to execute subsequent instructions irrespective of the first and second signals, in parallel to the execution of the instruction by the coprocessor.

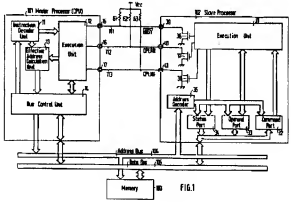


Fig.1

EP 0 313 097 A3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number

EP 88 11 7683

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.4)
A	EP-A-0 226 226 (WANG LABORATORIES INC.) 24 June 1987 " abstract; column 3, line 16 - column 4, line 22; column 4, line 56 - column 5, line 7; column 25, line 1 - column 27, line 25; column 28, lines 42-56; column 29, line 20 - column 30, line 17; column 32, lines 23-38; claims "	1-4	G06F9/38
A	EP-A-0 147 599 (INTERNATIONAL BUSINESS MACHINES CORP.) 10 July 1987 " page 3, lines 13-25; page 5, lines 11-24; page 6, line 33 - page 6, line 4; page 7, line 26 - page 8, line 6 "	1-4	
A	NEW ELECTRONICS, INCORPORATING ELECTRONICS TODAY, vol. 18, no. 19, October 1985, LONDON GB pages 66 - 72; J.-C. MATHON: 'Interfacing the 32081 as a floating point peripheral' " page 66, left column, lines 8-14; page 69, centre column, line 6 - right column, line 8; page 70, left column, lines 10-26; page 70, right column, line 4 - page 72, left column, line 32 "	1-4	TECHNICAL FIELDS SEARCHED (Int. Cl.4) G06F
A	WESCON 86/ Conference Record, vol. 30, 18-20 Nov 1986, Los Angeles, CA, US, P.A.Berger, "The National NS32381 Floating Point Processor", p35/2 1-6 " page 5, line 1 - line 30; figure 3 "	1	
A	WESCON 86/ Conference Record, vol. 29, 19-22 Nov 1986, San Francisco, CA, US, B.Beims, " The MC68020 32-bit MPU: Opening new application doors", p1/4 1-17 PAGE 9, right column, line 9 - page 10, left column, line 21 "	1	

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	08 JANUARY 1992	WEINBERG L.F.

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P0401)